Discovering Recurring Anomalies in Text Reports Regarding Complex Space Systems

Ashok N. Srivastava and Brett Zane-Ulman

Abstract—Many existing complex space systems have a significant amount of historical maintenance and problem data bases that are stored in unstructured text forms. The problem that we address in this paper is the discovery of recurring anomalies and relationships between problem reports that may indicate larger systemic problems. We will illustrate our techniques on data from discrepancy reports regarding software anomalies in the Space Shuttle. These free text reports are written by a number of different people, thus the emphasis and wording vary considerably.

We test four automatic methods of anomaly detection in text that are popular in the current literature on text mining. The first method that we describe is k-means or Gaussian mixture model and its application to the term-document matrix. The second method is the Sammon nonlinear map, which projects high dimensional document vectors into two dimensions for visualization and clustering purposes. The third method is based on an analysis of the results of applying a new clustering method, Expectation Maximization on a mixture of von Mises Fisher distributions, that represents each document as a point on a high dimensional sphere. In this space, we perform clustering to obtain sets of similar documents. The results are derived from a new method known as spectral clustering, where vectors from the term-document matrix are embedded in a high dimensional space for clustering.

The paper concludes with recommendations regarding the development of an operational text mining system for analysis of problem reports that arise from complex space systems. We also contrast such systems with general purpose text mining systems, illustrating the areas in which this system needs to be specified for the space domain.

TABLE OF CONTENTS

A. N. Srivastava, Ph.D. is at the NASA Ames Research Center (ashok@email.arc.nasa.gov). B. Zane-Ulman is with the Computer Sciences Corporation at NASA Ames (zane@email.arc.nasa.gov). This paper was published in the Proceedings of the 2005 IEEE Aerospace Conference.

Paper Number: 1241 version 2
0-7803-8870-4/05$20.00

1. INTRODUCTION

Many complex aerospace systems have a variety of prognostic and diagnostic instrumentation that deliver high speed data streams of information regarding the current health of the system. These streams give instantaneous information about the system and must be analyzed accordingly.

Along with these data streams, however, aerospace systems also have significant maintenance records associated with them. These maintenance records are often free-text reports. They are often recorded by maintenance personnel or engineers that are responsible for specific subsystems in the vehicle. In some cases, such as the Aviation Safety Reporting System [1], the reports are augmented by some structured data through the use of a coded report. The coded reports can be analyzed using standard statistical methods or data mining methods that are suited for the analysis of structured information.

The free-text reports, however, need to be significantly transformed to be analyzed with standard data mining or statistical methods. Most of those methods assume that the data can be expressed as a matrix where each row is an observation and each column is a variable. For example, in the case of analyzing the variations in reliability for 1000 different thermal sensors, a matrix could be formed which would have 1000 rows, and columns corresponding to various reliability metrics as well as other information regarding the sensors that are deemed relevant by the analyst. This information could include, for example, where the sensor was manufactured, when it was manufactured, information regarding the manufacturing process, etc. These pieces of information would form the columns of the data matrix that could then be submitted to a statistical or data mining analysis.

This paper discusses methods of analyzing free text documents where the text is represented in a matrix as described above– each document corresponds to a row in the matrix, and the columns correspond to the union of all the key words in all the documents. The entries in the matrix (called a term-document matrix) correspond to the frequencies of each key

word (or term) in the document. Through this procedure each document is represented by a point in a high dimensional vector space. This representation is used by many text analysis methods under the terms 'bag-of-words', latent semantic analysis, and other research areas [2]. A significant drawback of this vector space approach is that all semantic and syntactic information in the document is lost.

In the next section, we discuss the particular problem that we use to demonstrate our methodology and describe various approaches to discovering recurring anomalies. At the end of each section we discuss our experimental results. The paper concludes with a set of requirements for a text mining system architecture and presents conclusions and areas of future work.

2. Discovering Recurring Anomalies

The problem that we address in this paper is as follows. Given a set of N documents, where each document is a free text English document that describes a problem, an observation, a treatment, a study, or some other aspect of the vehicle, automatically identify a set of potential recurring anomalies in the reports. Note that for many applications, $N \approx 100,000$, which is a corpus that is too large for a single person to read, understand, and analyze by hand. Thus, while engineers and technicians can and do read and analyze all documents that are relevant to their specific subsystem, it is possible that other documents, which are not directly related to their subsystem still discuss problems in the subsystem. While these issues could be addressed to some degree with the addition of structured data, it is unlikely that all such relationships would be captured in the structured data. Therefore, we need to develop methods to uncover recurring anomalies that may be buried in these large text stores.

One approach to discovering recurring anomalies would be to develop a method to query the text database for known anomalies. For example, one could envision generating a list of queries, such as "find all examples of software errors", or "find all examples of navigation system faults", etc. While such a query mechanism is useful, it still does not address the problem of finding anomalies that may not be thought of a priori. The approaches that we describe in this paper are particularly useful for identifying unknown recurring anomalies.

The methods that we use to discover these anomalies are based on various clustering methods. *Clustering* refers to the process of identifying subsets of rows in the term-document matrix that have similar characteristics. The first approach that we discuss is based on the k-means clustering algorithm of the term-document matrix which implicitly makes Gaussian assumptions and uses the Euclidean distance between term-document vectors as a measure of similarity. The second clustering method uses the cosine measurement between two vectors and which implicitly assumes the von Mises Fisher distribution. The third clustering method, based on spectral clustering, embeds the term-document vectors in an infinite dimensional space and looks at the clustering of a low dimensional projection. These formulations will be discussed in the next section.

Our procedure for identifying recurring anomalies is based on the idea that similar anomalies will show up in the same cluster, and thus is highly dependent on the clustering algorithm. In this section, we describe three methods of cluster analysis that are popular in the literature and discuss their underlying assumptions. These assumptions affect the outcome of the clustering and therefore can affect the discovery of recurring anomalies.

For purposes of the discussion presented here, we will model the text as a term-document matrix [3]. The term-document is described by an $N \times p$ matrix Z, where N is the number of documents, and p is the number of keywords in the union of all documents. A keyword is defined as a word that is informative about the content of the document. Words such as 'and', 'the', 'but', and 'not' are called stop words and are abandoned when the term-document matrix is created. In many applications, $p \gg N$. In order to remove terms from this matrix that have small frequencies as compared to the number of documents, it is customary to perform a data reduction technique known as *Term Frequency Inverse Document Frequency* (TFIDF) to the term document matrix. We follow the notation in [3] as follows. For Z_{ij}, which corresponds to the entry in the matrix for the ith document d_i and the jth term t_j, TFIDF is a straightforward procedure and can be computed as follows:

$$Z_{ij} = TF(t_j, d_i) \times IDF(t_j) \tag{1}$$

$TF(t_j, d_i)$ is the term frequency, which is the frequency that term t_j appears in document d_i. $IDF(t_j)$ is the Inverse Document Frequency of term t_j and is defined as:

$$IDF(t_j) = \log(\frac{N}{DF(t_j)}) \tag{2}$$

where $DF(t_j)$ is the number of documents in the corpus that contain term t_j. Notice that if this number is close to N, the total number of documents in the corpus, $IDF(t_j) \approx 0$, and the term's contribution to the matrix is very small.

3. Data Description

The analysis we performed was based on a set of Flight Readiness Reports and Discrepancy Reports for the space shuttle. We received 358 Discrepancy Reports in pdf format, some contained text and some were scanned images. The Discrepancy Reports describe problems in the Space Shuttle software. They are a sample of such reports ranging in time from 1975 to 2000. The problems have to do with software issues across all shuttle subsystems.

We also received 35 Flight Readiness Reports. These are documents that are prepared before each mission and must be signed off before the shuttle is allowed to fly. Each one

describes problems that have occurred on previous missions that could affect the current mission. In the document, each of these problems is detailed in a separate section, called an Observation. For each observation a problem is described along with how it could affect the mission and what was done to correct the problem or a reason why it was considered an acceptable level of risk to ignore the problem. We separated these reports into 125 observations, which were treated as independent from one another. Because of the sensitive nature of the data, we cannot reveal the actual anomalies that were discovered, but can report on the success of algorithms on identifying anomalies.

4. DIMENSIONALITY REDUCTION

The TFIDF procedure outlined above can significantly reduce the number of dimensions (i.e., the number of columns) in the term document matrix. Our studies show that the reduction can be as much as 50-70% depending on the domain. However, in many cases it is necessary to reduce the dimension of the data even further. Principal Components Analysis (PCA) is an often used procedure for dimensionality reduction because of its simplicity and interpretability [4], [2].

While PCA has many advantages, it can suffer from the fact that only the linear structure in the data is preserved. There are many methods to perform nonlinear PCA using neural networks or other nonlinear learning algorithms, but the discussion of those algorithms is outside of the scope of this article [5].

PCA identifies the directions of maximum variation in the group of points defined by the document vectors. These directions can be shown to be the eigenvectors of the covariance matrix generated by the term-document matrix. Once the top l eigenvectors are identified (these correspond to those with the largest l eigenvalues), the document vectors are left-multiplied with the eigenvectors. This results in an l dimensional representation of the document vectors, where $l < p$. The parameter l is chosen in order to explain the maximum amount of variation in the data with the minimum number of eigenvectors. In the studies we describe here, PCA was used to reduce the dimensionality of the data. We demonstrate the effect of PCA on clustering and the identification of recurring anomalies.

5. k-MEANS ALGORITHM AND MIXTURE MODELS

The k-means clustering algorithm [6] is perhaps the most popular method of clustering structured data due to its simplicity of implementation. The algorithm works by choosing k random initial cluster centers, computing the distances between these cluster centers and each row in the data matrix and then identifying those rows that are closest to each cluster center. The corresponding cluster centers are moved to the centroid of those data points and the procedure is repeated. The algorithm converges when the cluster centers do not move from one iteration to the next.

The k-means algorithm is a special implementation of the Gaussian Mixture Model. These models assume that the data vectors are generated according to the probability density $P(Z_i|\Theta)$:

$$P(Z_i|\Theta) = \sum_{c=1}^{C} P(c)P(Z_i|\theta_c) \tag{3}$$

where Θ is a vector containing the C model parameters, and θ_c are the model parameters for the cth mixture component. The vector Z_i is a p dimensional vector from the term-document matrix. The parameters of this model are obtained through Expectation Maximization of the appropriate log-likelihood function or, more generally, the posterior log-likelihood. In the case of a Gaussian mixture density model for $Z_i \in \mathcal{R}^d$, we take the likelihood function as:

$$\begin{aligned} P(Z_i|\theta_c) &= P(Z_i|\mu_c, \Sigma_c, c) \\ &= (2\pi)^{-\frac{d}{2}}|\Sigma_c|^{-\frac{1}{2}} \times \\ &\quad \exp[-\frac{1}{2}(Z_i - \mu_c)^T \mathbf{\Sigma}_c^{-1}(Z_i - \mu_c)] \end{aligned}$$

Maximum a posteriori estimation is performed by taking the log of the posterior likelihood of each data point Z_i given the model Θ using the Expectation Maximization algorithm [7].

In the case of text clustering the vectors are high dimensional and sparse. Under these conditions, the k-means algorithm does not work well because the number of data points needed to form dense regions increases exponentially with the number of dimensions. The underlying assumption of the k-means algorithm is that there are dense regions in the data. With a finite amount of data and high dimension, most data points end up being approximately equidistant to each other.

k-means Results on Flight Readiness Review Data

We applied the k-means algorithm with $k = 7$ to the data from the Space Shuttle's Flight Readiness Reviews (FRR). The value of k was determined using cross-validation. The data was reduced to 30 dimensions using PCA and then clustered until the algorithm converged. Analysis of the results indicated that the FRR observations as classified by a human had a relatively low correlation with the results from the algorithm. Human classification of these documents revealed 15 different system level categories, such as "Alignment", "Contamination," "Design," etc. The distribution of documents across these 15 categories is shown in Figure 1 (upper panel). The middle panel of this figure shows the distribution of documents into clusters identified by the k-means algorithm.

The k-means algorithm also occasionally divided reports that were very similar (with small wording changes between them) into different clusters. This behavior was also noted when humans clustered the same documents. Other algorithms did not suffer from this difficulty.

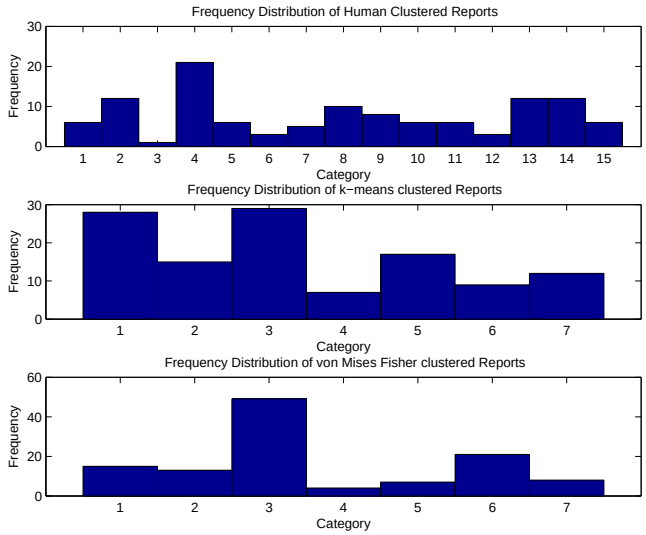


Figure 1. Upper panel: Distribution of documents across different clusters where a human read each document and manually clustered them into 15 different categories. Middle panel: distribution of documents in clusters using k-means. Lower panel: distribution of documents in clusters using von Mises Fisher clustering (see section 7).

6. SAMMON NONLINEAR MAPPINGS

The k-means algorithm clusters document vectors in the space of term frequencies and requires that the user determine the value of k. Thus, if the true number of clusters is greater than the predetermined value of k, those clusters would not be separately identified.

Sammon nonlinear maps are actually a method of projecting high dimensional data into a two or three dimensional plot for viewing and analysis purposes [8]. However, the maps can be quite helpful for visualization of recurring anomalies because the user does not define a number of clusters. Instead, the map is generated along with a quality of fit metric that can be visually inspected for recurring anomalies. Documents that appear close together in the map but are far away from the dense regions in the map have similar characteristics to each other but are different from the 'typical' document in the corpus. Therefore, these are candidate recurring anomalies and should be carefully reviewed to determine whether they are in fact recurring anomalies or whether they are different versions of the same reports.

In the systems that we are analyzing, there can be multiple versions of the same report. Since the number of reports can be very large and the version number is not clearly identified, it is not easy to identify nearly identical documents in the corpus. These documents are readily identified using this method.

Sammon nonlinear maps work by creating a two dimensional map that approximates the inter-point distances in the original high dimensional space. Thus, with N data points Z_i that are embedded in an l dimensional space, we generate a set of new points, Y_i in a two dimensional space such that

$$\sum_i \sum_j ||d(Z_i, Z_j) - d(Y_i, Y_j)||^2 \qquad (4)$$

is minimum, where d measures the Euclidean distance between real vectors. This problem is solved using standard gradient descent methods and yields interesting results when applied to text documents. This will be discussed in a subsequent section. Note that equation (4) is not exactly the cost function that is minimized in the Sammon Map.

Our studies indicate that for text documents, application of the algorithm directly to the p dimensional data (i.e., the document vectors that arise after the TFIDF procedure) may not yield good results. We perform PCA to make an initial dimensionality reduction and then use the Sammon map on the resulting data set [1].

Results of Sammon Maps on Flight Readiness Review Data

We applied the Sammon Map to the FRR data using two approaches. In the first approach, we directly mapped the data from the original high dimensional space down to two dimensions. Figure 2 shows the results of this mapping in terms of the sorted inter-document distances. The intuition behind these plots is as follows: the sorted distances of the documents in the original space should be very close to the sorted distances of the points in the two dimensional map. If they were identical, the lines would perfectly overlap. The top panel shows that there is significant error in the mapping at close distances (left hand side of the plot) as well as at far distances.

The middle panel shows the effect of linear dimensionality reduction using PCA followed by Sammon mapping. Here, the two curves almost completely overlap each other. The map generated using this procedure is shown in Figure 3.

One issue that arises with the Sammon map is that it generates an inter-document distance matrix of size $N \times N$. In the examples given here, N is relatively small, so these computations are easily performed on a desktop computer. However, for very large document corpora, this procedure cannot be implemented directly. Thus, we investigated the possibility of learning the Sammon map using a neural network and then using the neural network to project future documents into the two dimensional map [9]. The results of this procedure are shown in the lower panel on test data. The original data set was broken into a training, validation, and test set. The test set was approximately 40% of the total data available. Notice that the neural network does a good job of mapping the documents that are close to each other, but makes errors with

[1] In some cases one can use the first two principal components as an approximation to the relative location of the data in the original space. However, the results are highly dependent on the relative sizes of the eigenvalues of the correlation matrix

mid-distance documents.

Figure 3 gives the Sammon map when dimensionality reduction is done using PCA. The contours indicate regions where recurring anomalies are likely. FRRs that are nearly identical are correctly placed close together in the visualization, such as in the lower left hand part of the graph. The large cluster on the right hand side of the graph shows potentially interesting recurring anomalies.

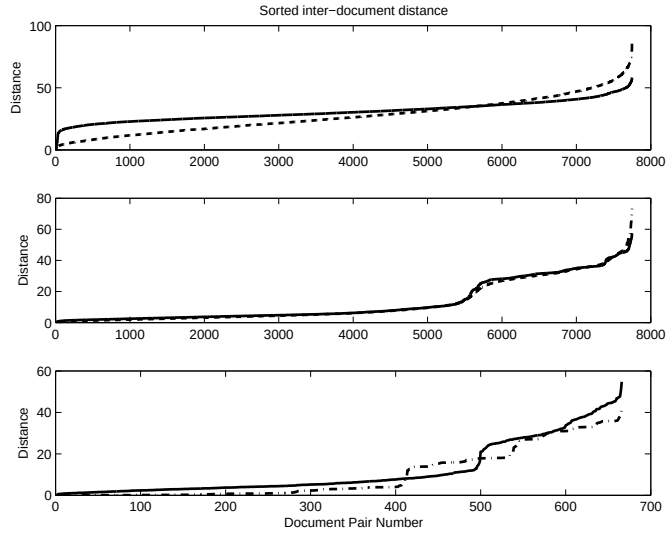


Figure 2. The top panel of this plot shows the sorted inter-document distances in the original 500 dimensional space and the distances that arise from a 2 dimensional approximation to the original distances. Original distances are shown in the solid line, and the dotted line shows the distances with the 2 dimensional approximation. Notice that there is substantial error in the approximation. The middle panel shows the results of Sammon mapping after the dimension of the document space is reduced from 500 dimensions to 10 dimensions using principal components analysis. The agreement between the distances in the low dimensional space and the 2 dimensional mapping are excellent. The bottom panel shows the approximation of the Sammon mapping using a neural network.

7. VON MISES FISHER CLUSTERING

The Gaussian Mixture Model and k-means algorithms make Gaussian assumptions about the underlying distribution of the data. Empirical studies have shown that for high dimensional sparse data sets, the cosine measure of similarity between two vectors is a better measure than the Euclidean distance. A recent paper [10] developed the mathematics to perform clustering using the cosine measure of similarity. Just as the Euclidean distance implicitly implies a Gaussian distribution the cosine distance implicitly implies a different distribution, known as the von Mises Fisher distribution. We follow the

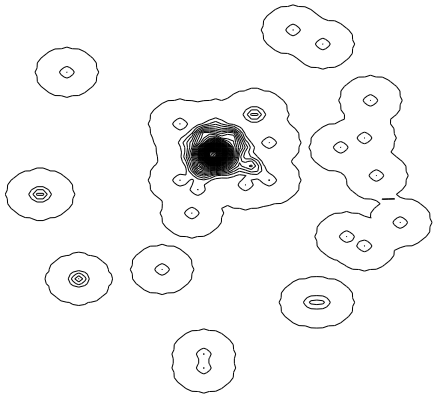


Figure 3. This visualization is a projection of the 500 dimensional document vectors into two dimensions using Sammon mapping. PCA is performed first to reduce the dimension of the data to 10 dimensions, and the Sammon map is generated from the lower dimensional data. The contours represent regions of equiprobability. Recurring anomalies can be documents that fall within the same closed contour.

formulation in [10] closely:

$$P(Z_i|\Theta) = \sum_{c=1}^{C} P(c)P(Z_i|\theta_c) \tag{5}$$

In this case, we assume that the vectors Z_i have been normalized to unit length. For p dimensional data vectors, we have the von Mises Fisher (vMF) distribution:

$$P(Z_i|\mu,\kappa) = c_p(\kappa)\exp(\kappa\mu^T Z_i) \tag{6}$$

where μ is a unit vector corresponding to the mean of the distribution and $\kappa \geq 0$ is the measure of dispersion. The constant $c_p(\kappa)$ is given by:

$$c_p(\kappa) = \frac{\kappa^{(p/2)-1}}{(2\pi)^{(p/2)} I_{(d/2-1)}(\kappa)} \tag{7}$$

where $I_{(r)}(\kappa)$ represents the modified Bessel function of the first kind of order r. With the vMF distribution as defined above, Banerjee et. al. (2003) derive the Expectation Maximization algorithm to optimize a mixture of vMF distributions [10]. Their results indicate that this algorithm has superior performance on high dimensional text clustering problems compared to the k-means algorithm.

Results of vMF Clustering

The vMF clustering method described here was applied to the Flight Readiness Reports. This clustering algorithm correctly clustered similar documents into the same cluster. As with k-means, the number of clusters needs to be chosen. We chose a value of $k = 7$ since that produced the best results on a cross-validation set. Figure 1(bottom panel) shows the distribution of the documents across clusters.

The clusters discovered by vMF clustering indicated several areas of interest to investigate for recurring anomalies in dif-

ferent subsystems. When comparing with the results from the human clustered documents, we found that the themes of the reports were well grouped using this algorithm, and that the document clusters assigned by hand were grouped appropriately within the vMF clusters.

For the Discrepancy Reports we also were given a set of groupings that was done by shuttle software team members. We were able to compare our own clustering results with these. We found that in several cases, documents that were very similar had been grouped in separate clusters by the software team members but were identified with the same cluster by our clustering software.

8. Spectral Clustering

Spectral clustering is a different approach to clustering that works by embedding the vectors Z_i in a high, possibly infinite dimensional space using Mercer Kernels [11]. Mercer Kernel functions can be viewed as a measure of the similarity. For a finite sample of data $\mathcal{Z}$, the kernel function yields a symmetric $N \times N$ positive definite matrix, where the (i, j) entry corresponds to the similarity between (Z_i, Z_j) as measured by the kernel function. Because of the positive definite property, such a Mercer Kernel can be written as the inner product of the data in the feature space. Thus, if $\Phi(Z_i) : \mathcal{R}^d \mapsto \mathcal{F}$ is the (perhaps implicitly) defined embedding function, we have $K(Z_i, Z_j) = \Phi(Z_i)\Phi^T(Z_j)$. Typical kernel functions include the Gaussian kernel for which $K(Z_i, Z_j) = \Phi(Z_i)\Phi^T(Z_j) = \exp(-\frac{1}{2\sigma^2}||Z_i - Z_j||^2)$, and the polynomial kernel $K(Z_i, Z_j) = \Phi(Z_i)\Phi^T(Z_j) =< Z_i, Z_j >^p$.

For supervised learning tasks, linear algorithms are used to define relationships between the target variable and the embedded features [12]. Work has also been done in using kernel methods for unsupervised learning tasks, such as kernel clustering [13], [14] and density estimation [15].

Spectral clustering works by computing the eigenvectors of a normalized kernel matrix (see [11] for details of the algorithm). The largest n eigenvectors are chosen and normalized to unit length. The rows of the eigenvectors (corresponding to N points in an n dimensional space) are then clustered using the k-means algorithm.

Results of Spectral Clustering

We performed clustering on the term-document matrix using spectral clustering and the Gaussian kernel $K(Z_i, Z_j) = \exp(-\frac{1}{2\sigma^2}||Z_i - Z_j||^2)$. This clustering procedure has three parameters associated with it: k, which is the number of clusters, the number of eigenvectors chosen (number of dimensions), and the scale parameter σ^2. The choice of these parameters significantly affects the quality of the results. We chose to measure the quality of the results by computing the dispersion of the data within the cluster. Higher dispersion means that the clusters are broader, and less well defined. We are interested in finding clusters with low dispersion. We scanned a portion of the three-dimensional parameter space and marginalized across the third parameter to show a two dimensional contour map of dispersion as shown in Figure 4. Based on these maps, we chose $k = 16$, number of dimensions = 11, and $\sigma^2 = 30$. The document frequencies are shown in Figure 5.

We explored the application of Sammon mapping to the results of the spectral decomposition of the kernel matrix as shown in Figure 6 to determine the geometry of the data that was clustered. This map clearly shows three well defined clusters and a dense region of documents. The clusters again identify similar reports as well as potentially interesting groups of anomalies. The highly dense region corresponds to cluster 8 in Figure 5.

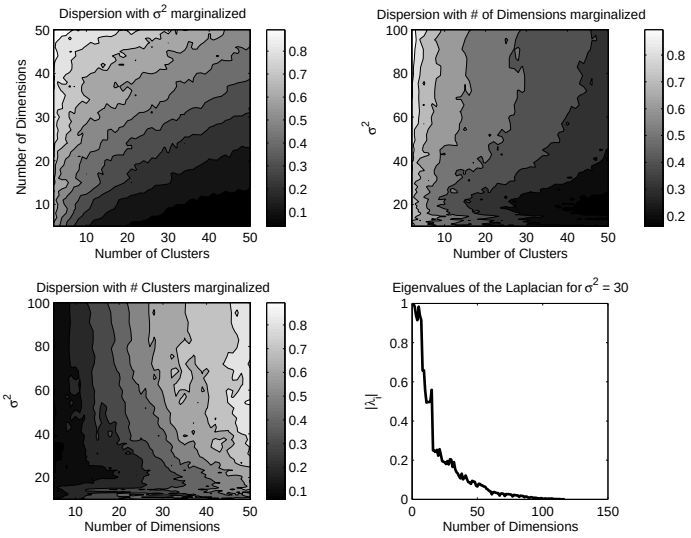


Figure 4. This visualization shows how the clustering results vary with the three parameters in spectral clustering: the number of dimensions, the number of clusters, and σ^2, which is the scale parameter in the kernel.

9. System Architecture for Discovering Recurring Anomalies

In this section we describe an architecture for a system that could manage data to help discover recurring anomalies. An aerospace vehicle is a highly complex system with complex interactions between its various subsystems. To get the most out of a problem tracking system as many of these complex relationships as possible need to be included in the tracking and analysis of issues that arise. The system we propose contains an engineering model of the vehicle detailing the relationship between vehicle components and subsystems. This model is joined to a relational database containing additional vehicle component information as well as structured fields for entering problem reports via forms. This information gives the system a better context in which to do clustering of the problem reports, thereby increasing the likelihood that meaningful clusters will be produced. The need for this type of organized, interconnected structure was found to be impor-

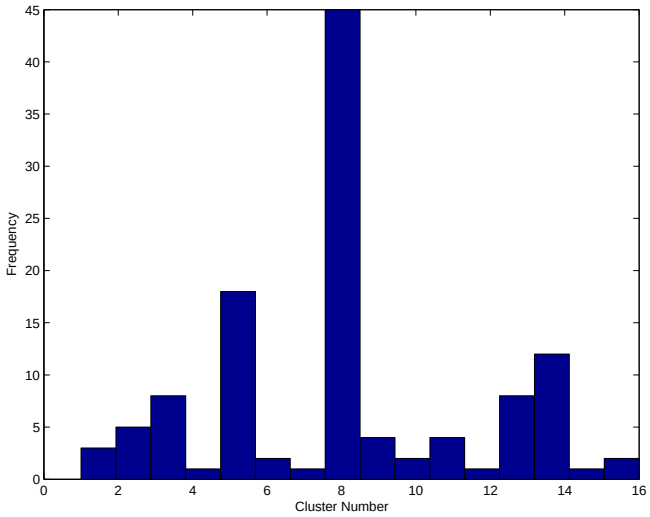


Figure 5. This diagram shows the frequency distributions of documents with clusters. The large cluster corresponds to the dense region in the previous figure.

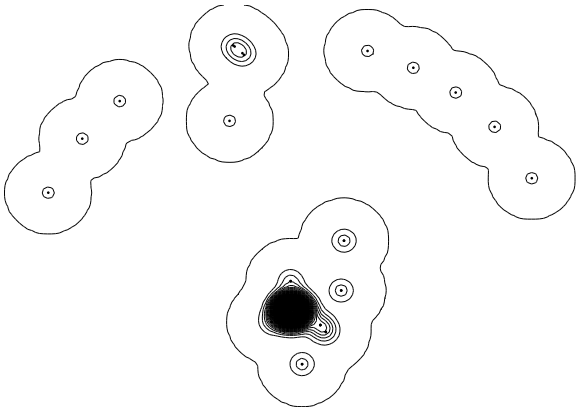


Figure 6. This visualization is a projection of the eigenvectors from the kernel matrix into two dimensions using Sammon mapping. The contours represent regions of equiprobability. Recurring anomalies can be documents that fall within the same closed contour.

tant in the NASA Space Shuttle program by an independent assessment team [16].

The vehicle model should consist of ontology of the language used to describe the vehicle and its components, domain information, and vehicle system structure.

The ontology portion defines the language of terms used when describing problems with the vehicle. This includes acronym definitions, thesaurus terms, conceptual hierarchies, and irrelevant terms. Commonly used acronyms need to be defined so that their terms can be related between documents with related, but not identical, references. A set of thesaurus terms will help to relate documents by their intended meaning, not just their literal content. Conceptual hierarchies

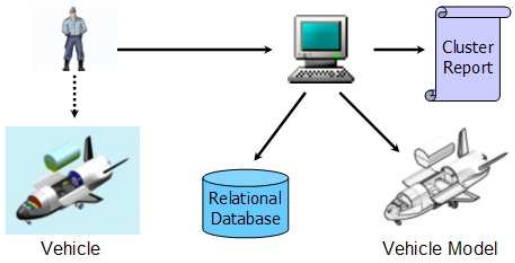


Figure 7. System Architecture - Engineers observing the vehicle enter problem reports into the system, which are stored in a relational database and joined with vehicle component and design information. This allows for flexible reporting and more relevant analysis of trends in the problem reports.

group sets of terms into low level concepts, and low level concepts into higher level concepts. It can be thought of as a tree structure, with all of the terms in the language as the leaves of the tree. The parent node of a set of terms is the concept shared by all of those terms. At the next level up the tree these low level concepts are joined by a parent node which groups them into a higher level concept. This continues up the tree (hierarchy) to the root node, which joins all the highest level concepts together and represents the base concept of the entire language. This hierarchical structure helps to put terms in context and to create links between documents. The weight of the links can be varied depending on the level in the conceptual hierarchy that the link was made. A set of irrelevant terms should also be included in the vehicle model to identify common terms or codes that shouldn't be used when clustering documents. These are also known as stop words, and include words such as 'the', 'and', 'to', etc., as well as common words specific to the domain.

The domain information consists of relationships between terms. Terms can be related by causality (ie. 'water' causes 'corrosion'), similarity, mutual exclusivity, etc. These relationships should describe physical and engineering relationships that are specific to the vehicle design.

The vehicle system structure is an engineering model that defines how parts, components, and subsystems interact with each other.

The relational database consists of tables for all of the vehicle parts, components, and subsystems. It also has transactional tables for entering problem reports with both fixed fields and free text fields. Setting up the database for problem tracking in this manner will allow for simple and complex queries to answer common high level questions as well as give a great deal more information to the clustering algorithms.

The part table should have a part ID as a primary key. Each record should be a unique part with fields describing properties of the part as well as component IDs of each component

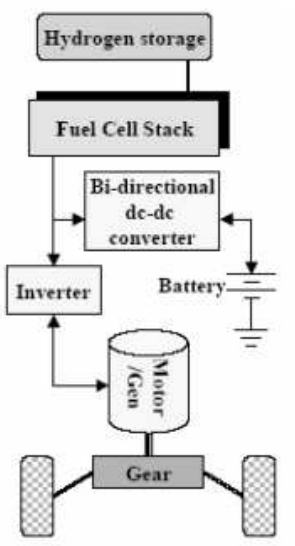


Figure 8. Example vehicle system structure - specifies how components and subsystems fit together so that analysis methods can take into account interactions between subsystems.

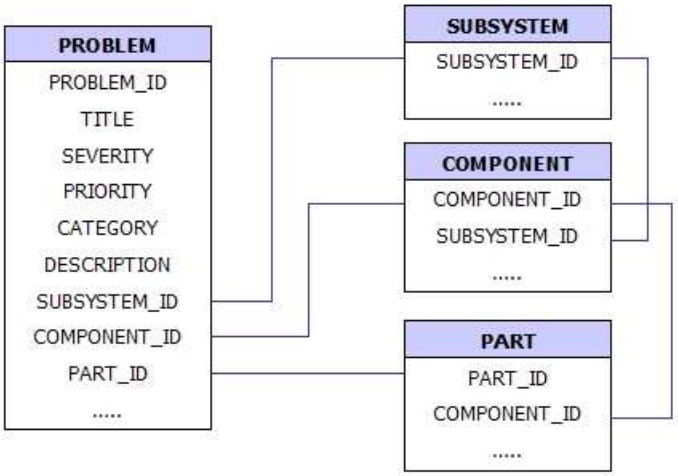


Figure 9. Relational Database Schema - joins problem reports with detailed information about the vehicle's parts, components, and subsystems.

the part is used for in the vehicle. These can be used for joining with the component table.

Similarly, the component table should have a component ID as the primary key and each record should describe properties of the component. It should list subsystem IDs of each subsystem in which the component is used.

The problem report tables are similar to bug tracking systems commonly used in software development. They should consist of fixed fields for things like title, priority, problem category, severity, and subsystem or component if applicable. The free text field is the main body of the problem report where a full description and discussion of the problem is entered.

The system can be used to perform clustering of problem reports in order to discover recurring anomalies. As observers discover problems on the vehicle they enter them into the relational database through a simple web interface. If the vehicle system design changes then the vehicle model is updated to reflect those changes. Regular reports of open issues can be generated simply by querying the database. A streamlined, efficient process flow for entering and analyzing problem reports is critical for analyzing trends and recurring anomalies [17]. To this end, the system architecture, clustering algorithms, and methods described above can be used.

10. Conclusions and Future Work

Our experimental results indicated that the k-means algorithm does not perform well in high dimensional spaces. This is understandable since k-means has an underlying assumption that there are dense regions in the data, and with a small number of data points in a high number of dimensions this will not be the case.

Sammon mapping was found to be an extremely useful way of clustering and visualizing high dimensional data points. The data are projected down to two dimensions such that the relative distance between any two points is maintained as closely as possible. Our experimental results showed that the distances in two dimensions matched the distances between the high dimensional points extremely well. This method is therefore well suited producing clusters that can then be analyzed by hand to investigate whether any particular cluster captured a set of recurring anomalies.

The results of our experimentation with the Expectation Maximization algorithm over a mixture of von Mises Fisher distributions also looked promising. The clusters found by the algorithm had a high correlation with clusters created by hand. This requires further investigation to take into account a larger set of documents and several human classifiers to account for the subjectivity of clustering by hand.

Spectral clustering appears to segment the data into well separated clusters. In conjunction with Sammon mapping this algorithm produced several small, separate clusters which can be investigated for recurring anomalies.

Our discussion of a system architecture may be of use to aerospace projects in their early stages. To get the most value from a problem tracking system it is best to design the system for analysis from the beginning. The methods discussed in this paper present a practical and general way to maximize the analytical value of the tracking system, and therefore the safety and reliability of the vehicle.

To make more conclusive statements about our results will

require future work. We will validate the methods described in this paper on larger data sets with and compare with several expert human clusterings. We also plan to incorporate semantic information to the analyses which can give the clustering algorithms the ability to link documents based on their conceptual content, not just the words they contain.

11. ACKNOWLEDGEMENTS

This work was supported by the NASA Aviation Safety Measurement and Modeling Program, the NASA Intelligent Systems Intelligent Data Understanding Program, and the NASA Engineering Safety Center. The authors would like to thank Bill Macready, Nikunj Oza, Smadar Shiffman, and Avik Sarkar for valuable feedback and discussion.

REFERENCES

[1] L. Connell, "Incident reporting: The nasa aviation safety reporting system," *GSE Today*, pp. 66–68, 1999.

[2] T. K. Landauer, D. Laham, and P. Foltz, "Learning human-like knowledge by singular value decomposition: A progress report," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10. The MIT Press, 1998. [Online]. Available: citeseer.ist.psu.edu/landauer98learning.html

[3] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," in *Proceedings of ICML-97, 14th International Conference on Machine Learning*, D. H. Fisher, Ed. Nashville, US: Morgan Kaufmann Publishers, San Francisco, US, 1997, pp. 143–151. [Online]. Available: citeseer.ist.psu.edu/joachims96probabilistic.html

[4] I. T. Jolliffe, *Principle Components Analysis*. New York: Springer-Verlag, 1986.

[5] B. Scholkopf, A. Smola, and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.

[6] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm, Tech. Rep. AIM-1440, 1993. [Online]. Available: citeseer.ist.psu.edu/article/jordan94hierarchical.html

[7] A. P. Dempster, M. Laird, N., and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society B*, 1977.

[8] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, pp. 401–409, 1969.

[9] D. de Ridder and R. Duin, "Sammon's mapping using neural networks: A comparison," 1997. [Online]. Available: citeseer.ist.psu.edu/ridder97sammons.html

[10] A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra, "Generative model-based clustering of directional data," 2003. [Online]. Available: citeseer.ist.psu.edu/article/banerjee03generative.html

[11] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," 2001. [Online]. Available: citeseer.ist.psu.edu/ng01spectral.html

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

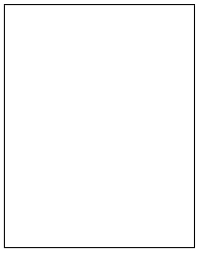
[13] M. Girolami, "Mercer kernel based clustering in feature space," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 780–784, 2001.

[14] A. N. Srivastava, "Mixture density mercer kernels: A method to learn kernels directly from data," *Proceedings of the 2004 SIAM Data Mining Conference*, 2004.

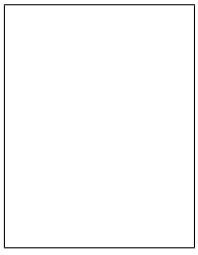
[15] W. G. Macready, "Density estimation with mercer kernels," *Technical Report TR03.13 of the Research Institute of Advanced Computer Science*, 2003.

[16] H. McDonald, "Shuttle independent assessment team report," Splace Shuttle Independent Assessment Team Report to Associate Administrator Office of Space Flight, Tech. Rep., 1999. [Online]. Available: www.hq.nasa.gov/osf/siat.pdf

[17] C. Linde and R. Wales, "Work process issues in nasa's problem reporting and corrective action (praca) database," NASA Ames Research Center, Human Factors Division, Tech. Rep., 2001. [Online]. Available: human-factors.arc.nasa.gov/april01-workshop/2pg-linde3.doc



Ashok N. Srivastava *is a Principal Scientist and Group Leader in the Data Mining and Complex Adaptive Systems Group at NASA Ames Research Center. He has sixteen years of research, development, and consulting experience in machine learning, data mining, and data analysis in time series analysis, signal processing, and applied physics. Specific applications in signal processing, text mining, and integrated system health management are addressed in his research.*



Brett Zane-Ulman *is a Computer Scientist at Computer Sciences Corporation (CSC), in the Data Mining and Complex Adaptive Systems Group at NASA Ames Research Center. He has seven years of software development and consulting experience in data mining and data visualization. He has developed enterprise software at SGI and Blue Martini Software.*